



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/065,793	11/19/2002	Jan Phillippe Eiras	1416.01	4753
21901	7590	11/01/2010		
SMITH HOPEN, PA 180 PINE AVENUE NORTH OLDSMAR, FL 34677			EXAMINER EL CHANTI, HUSSEIN A	
			ART UNIT 2441	PAPER NUMBER
			NOTIFICATION DATE 11/01/2010	DELIVERY MODE ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PATENTS@SMITHHOPEN.COM
ajhopen@yahoo.com
PAIR@SMITHHOPEN.COM

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte JAN PHILLIPPE EIRAS and MARTIN RIPPER

Appeal 2009-006328
Application 10/065,793
Technology Center 2400

Before ROBERT E. NAPPI, JOHN C. MARTIN, and
JOSEPH F. RUGGIERO, *Administrative Patent Judges*.

MARTIN, *Administrative Patent Judge*.

DECISION ON APPEAL¹

¹ The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the “MAIL DATE” (paper delivery mode) or the “NOTIFICATION DATE” (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

STATEMENT OF THE CASE

This is an appeal under 35 U.S.C. § 134(a) from the Examiner's rejection of claims 1-31, which are all of the pending claims.²

We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

A. Appellants' Invention

Appellants' invention is a low-cost, customizable system for non-intrusively testing the interaction of interfaces amongst components of nearly any system or systems. Specification [0008]. The invention accomplishes this by simulating the components' expected interfaces and acting as a communications "wire tap," capturing and processing real-time message traffic, and then analyzing and acting on the results (*id.*).

Appellants' Figure 3 is reproduced below.

² The failure of the Final Action to identify claim 31 as a rejected claim was remedied by the June 19, 2008, Office communication.

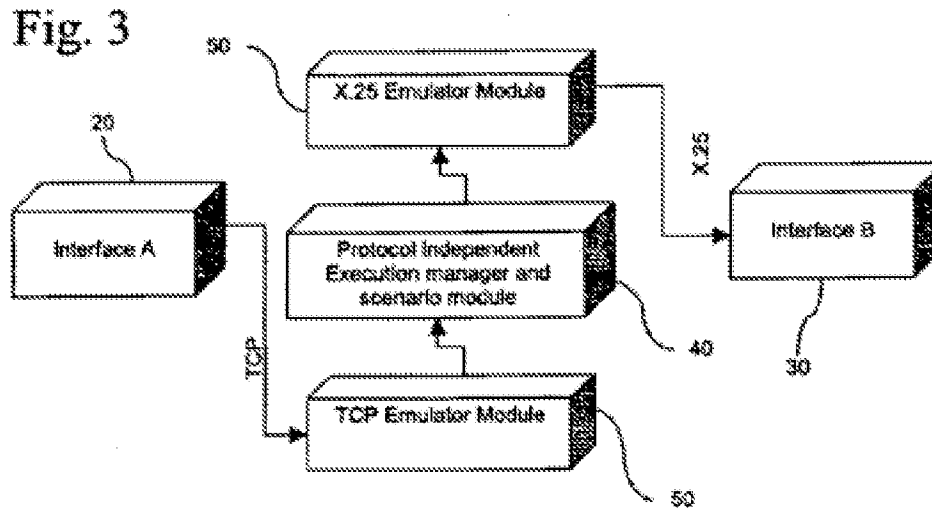


Figure 3 is a diagrammatic view showing the invention being used with an Interface A (20) that communicates using the TCP protocol and an Interface B (30) that communicates using the X.25 protocol (*id.* at [0022], [0047]). These interfaces are communicatively coupled via an X.25 emulator module and a TCP emulator module (each 50) to an execution manager and scenario module 40 (*id.*), which is also referred to in the Specification (at [0016]) as “the API core” and in claim 1 as “a protocol independent API core module.” Messages received from outside by the emulators are reformatted into scenario-compatible messages and handed to the execution manager, and outgoing messages accepted from the execution manager are reformatted as appropriate for the external interfaces (*id.* at [0050]).

B. Claim 1

Claim 1, the sole independent claim, reads as follows:

1. A computer program product for program level message traffic interception comprising:
 - a computer-readable medium;
 - a protocol independent API core module stored on the medium, the API core module having an array of predetermined rules for intercepted message traffic; and
 - an interface communication emulator module communicatively coupling protocol-specific program level message traffic to the API core.

Claims App. (Br. 12).

C. The reference

The rejection is based on:

Hite	US 6,763,040 B1	July 13, 2004
------	-----------------	---------------

D. The rejection

Claims 1-31 stand rejected under 35 U.S.C. § 102(e) for anticipation by Hite. Final Action 2, para. 2.

THE ISSUES³

Appellants agree that Hite fails to anticipate claim 1 in three respects.

³ See *Ex parte Frye*, 94 USPQ2d 1072, 1075 (BPAI 2010) (precedential) (“If an appellant fails to present arguments on a particular issue — or, more broadly, on a particular rejection — the Board will not, as a general matter, unilaterally review those uncontested aspects of the rejection.”). Designated precedential at
(Continued on next page.)

Appellants also argue that the finality of the Final Action is premature and should be withdrawn (Br. 9). This argument is entitled to no consideration because it concerns a matter that is petitionable rather than appealable. MPEP § 1002.02(c), para. 3(a) (rev. July 2008).

ANALYSIS

Hite's invention provides Internet control of a control area network (CAN) that includes, for example, appliances such as heating, ventilation, and air conditioning (HVAC) systems, lighting systems, audio-visual systems, telecommunications systems, security systems, surveillance systems, and fire protection systems. Hite, col. 1, ll. 15-21, 30-33. In one aspect of the invention, the boundaries between the Internet and the CAN are made transparent and the Internet becomes a device on the CAN (col. 1, ll. 34-37).

Figure 2 is reproduced below.

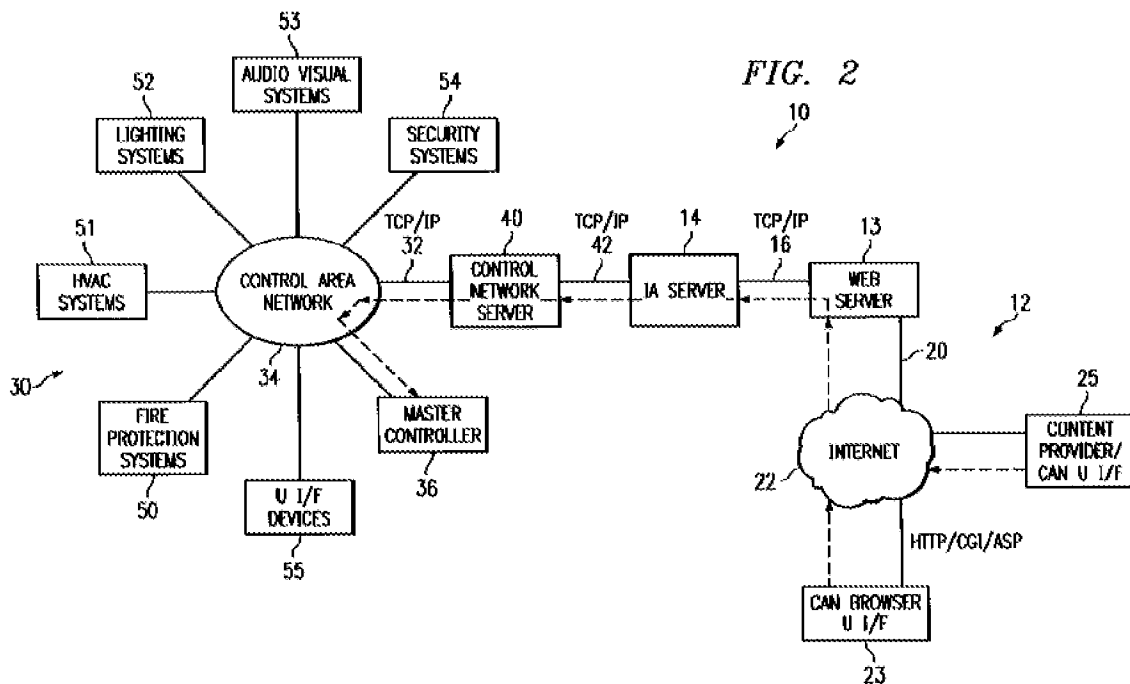


Figure 2 is a detailed block diagram of a system 10 coupling one or more control systems to the Internet in accordance with an embodiment of Hite's invention (col. 5, ll. 4-7). CAN portal 12 includes a web server 13 that couples the Internet 22 to an Internet appliance (IA) server 14 (col. 5, ll. 7-10). IA server 14 is coupled in turn to a control network server 40, which in turn is coupled to a CAN 30 that includes a master controller 36, user interface devices 55, and a plurality of appliances and systems, such as fire protection systems 50 (col. 5, ll. 10-16).

Figure 3 is reproduced below.

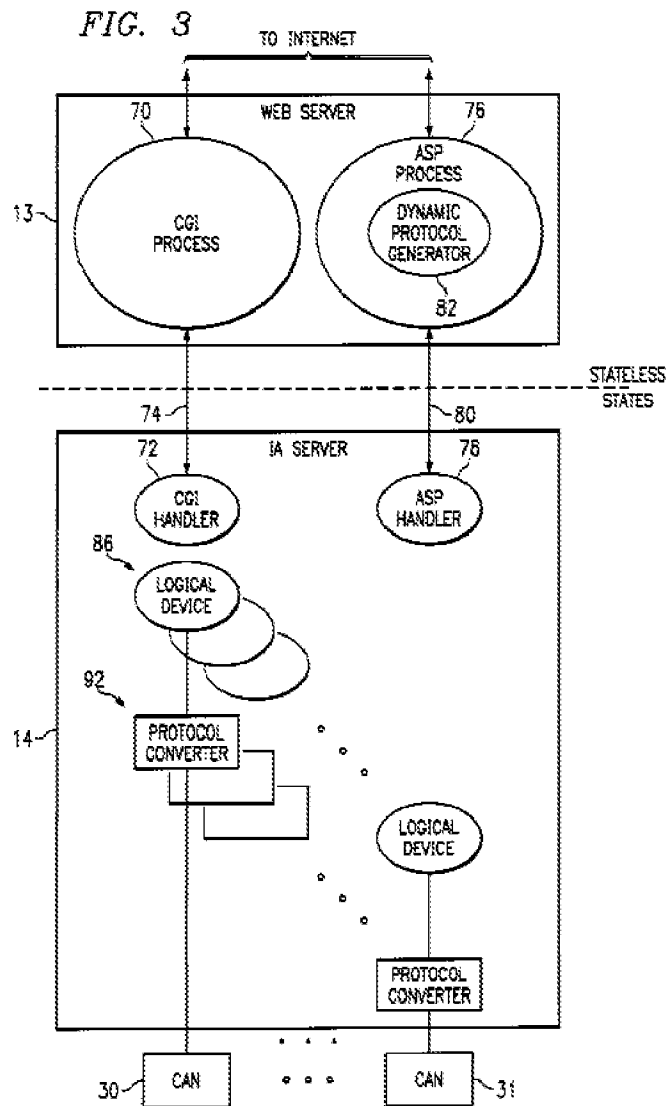


Figure 3 is a detailed block diagram of the processes in, and communications between, web server 13 and an IA server 14 (col. 5, ll. 38-42). Web server 13 can include (a) one or more CGI (Common Gateway Interface⁴) processes 70 for responding to CGI requests from the Internet

⁴ Hite, col. 3, l. 62.

and (b) one or more ASP (Microsoft® Active Server Pages⁵) processes 76 for responding to ASP requests from the Internet (col. 5, ll. 42-45).

IA server 14 includes a CGI handler 72 and an ASP handler 78, each of which provides a translation function from an IP protocol to a protocol used in the CANs, such as PhastLink or AXLink (col. 6, ll. 1-4). One or more protocol converters 92 can be provided to translate from the protocol used internally in the IA server, such as ICSP (Internet control system protocol⁶), to other protocols used in the CANs, such as AxLink or PhastLink (col. 6, ll. 13-16). However, a protocol converter is not necessary if the protocols employed in IA server 14 and CAN 30 are the same (col. 6, ll. 16-18).

Figure 4 is reproduced below.

⁵ Hite, col. 3, ll. 61-62.

⁶ Hite, col. 5, l. 52.

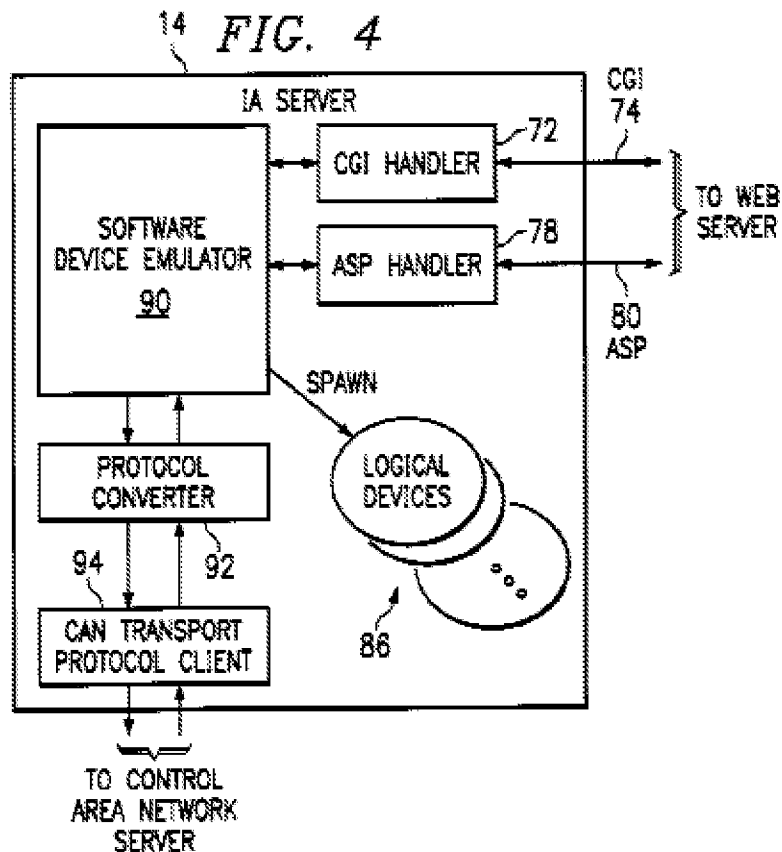


Figure 4 is a more detailed block diagram of the IA server processes for coupling one or more control systems to the Internet (col. 6, ll. 19-22). Software device emulator 90 spawns one or more software logical devices 86, which are software representations of devices connected to the CANs or content providers connected to the Internet (col. 6, ll. 23-27). Software device emulator 90 communicates with a protocol converter layer 92, which provides a protocol translation function between the IA server protocol and the CAN protocol, if they are different (col. 6, ll. 27-30). A CAN transport protocol client 94 is also provided to communicate with the CAN (col. 6, ll. 30-32).

Figure 11 is reproduced below.

	FIELD	SIZE (BYTES)
670	PROTOCOL FIELD	1
672	LENGTH OF DATA	2
674	FLAGS	1
676	DESTINATION SYSTEM	2
678	DESTINATION DEVICE	2
680	DESTINATION PORT	2
682	SOURCE SYSTEM	2
684	SOURCE DEVICE	2
686	SOURCE PORT	2
688	ALLOWED HOP COUNT	1
690	MESSAGE ID	2
692	MESSAGE COMMAND	2
694	MESSAGE DATA	4
696	CHECKSUM	1

FIG. 11

Figure 11 is a diagram of an exemplary packet for messages in Hite's system (col. 11, ll. 38-39). The packet includes, *inter alia*, a protocol field 670 identifying the format of the data section of the packet, a message command field 692, and a message data 694 field (col. 11, ll. 39-42; col. 12, ll. 31-32).

Table A, shown in columns 12-17, lists exemplary messages that are valid between a device manager and the device or master (col. 12, ll. 38-39).

In the Final Action, the Examiner provides the following explanation of how the first two paragraphs of claim 1 read on Hite:

a computer-readable medium (see col. 6 lines 4-47);

a protocol independent API core module stored on the medium, the API core module having an array of predetermined rules for intercepted message traffic (see col. 6 lines 48-67 and TABLE A, API core receives messages and are processed according to the instructions included using TABLE A)[.]

Final Action 2, para. 2. Appellants responded to this aspect of the rejection with two arguments. The first argument is that

[t]he API described by Hite is not protocol independent. As described by Hite at col. 1 and col. 11, a communication protocol is provided comprising a packet protocol having a protocol field for indicating the type of protocol, a length of data field for listing the length in bytes of the data field, a data field containing sub protocol data, and a checksum for determining the integrity of the packet. As such, the API described by Hite et al. is not protocol independent, but instead is dependent upon the specific protocol dictated by the internet appliance or the control area network selected.

(Br. 10-11.) The Examiner responded by stating that Hite's API is "protocol independent" because it is not limited to a *specific* protocol:

[T]he mere fact that the messages include protocol information does not make the API taught by Hite dependent on a specific protocol. . . . [T]he IA server includes protocol converters 92 to convert messages between the CAN server and the client (see fig. 3 and 4 and col. 6 lines 19-32) and therefore the API taught by Hite is not dependent on a specific protocol since the software device core is capable of handling a plurality of protocols.

(Answer 9.) As support for this claim interpretation, which allows the API to have a protocol, the Examiner explains that

the claim language used by the applicant defines the claimed "protocol independent API" to have "an array of predetermined rules for intercepted message traffic". Protocol in the simplest

definition is a *set of rules*. Therefore the claimed API is dependent on an array of rules which renders the claimed API dependent on some form of protocol. Even assuming that Hite's API processes protocol information, according to the definition of the claimed protocol independent API, Hite still teaches the scope of the claimed limitation.

(Answer 9.) Appellants, who did not file a Reply Brief, have not identified any alleged error in this claim interpretation.⁷

Appellants' second argument is that the Examiner erred in reading the recited "array of predetermined rules" on Table A. Specifically, Appellants contend that "TABLE A is a list of exemplary messages that are valid between a device manager and a device master. These exemplary valid messages are not equivalent to the array of predetermined rules for intercepted message traffic as disclosed and claimed by the present invention." (Br. 11.) The Examiner responded to this argument by further finding that Hite

teaches the software device emulator 90 also includes a CAN input message processor 102 that receives input messages and parses fields of the message [to] determine a message destination. Software device emulator also processes the messages to

⁷ An appellant may attempt to overcome an examiner's obviousness rejection on appeal to the Board by: (A) submitting arguments and/or evidence to show that the examiner made an error in either (1) an underlying finding of fact upon which the final conclusion of obviousness was based or (2) the reasoning used to reach the legal conclusion of obviousness; or (B) showing that the prima facie case has been rebutted by evidence of secondary considerations of nonobviousness. *Ex parte Frye*, 94 USPQ2d 1072, 1075 (BPAI 2010) (precedential)

determine a current state of a software logical device and a next state in response to the processed message (see col. 6 lines 44-67). At least the determination of the destination of messages and the state of the logical device are interpreted by [the] examiner to be “an array of predetermined rules”. [The c]laim language does not specify what the contents of the array of rules are and therefore [the] examiner broadly interprets the parsing of the message and determining the destination and the state of the device to be the array of predetermined rules.

(Answer 10.) Appellants have not asserted any error in this reasoning by the Examiner.

The Examiner reads the last paragraph of claim 1 on Hite as follows: “an interface communication emulator module communicatively coupling protocol-specific message traffic to the API core (see col. 1 lines 55-58, the received messages are provided with program specific protocol).” Final Action 3. These cited lines in column 1 describe the function of dynamic protocol generator 82 in ASP process 76 in web server 13 (Fig. 3). As explained in column 5, lines 61-65, the function of this dynamic protocol generator is to enable a scripting language such as VBScript or JavaScript to directly communicate on any TCP/IP network connection. These scripting languages are also discussed at column 51, lines 41-46.

Appellants, after summarizing Hite’s above-noted description of dynamic protocol generator 82, argue that “Hite does not describe an interface communication emulator module that handles the actual receipt and transmission of messages on a specific type of interface as disclosed and

claimed by the present invention.” (Br. 11.) The Examiner responded to this argument by additionally finding that

Hite discloses protocol converter 92 and CAN transport protocol client 94. The CAN transport protocol client 94 receives messages from the CAN server. The protocol converter converts messages between the CAN server and the client (see fig. 3 and 4 and col. 6 lines 19-32). The messages are then forwarded to the software device emulator to be transmitted through the API to the web server (see fig. 3 and 4). Therefore protocol converter 92 and CAN transport protocol client 94 are interpreted to be “an interface communication emulator module” that receives protocol messages from the CAN server and is forwarded to the API 110 and therefore “coupling the message traffic to the API”.

(Answer 10-11.) Appellants also have not identified any alleged error in this reasoning.

Because Appellants have not persuaded us that the Examiner erred in rejecting claim 1 for anticipation by Hite, we are sustaining the rejection of claim 1 and the rejection on that ground of dependent claims 2-31, which are not separately argued. *In re Nielson*, 816 F.2d 1567, 1572 (Fed. Cir. 1987).

DECISION

The Examiner’s decision that claims 1-31 are unpatentable under 35 U.S.C. § 102(e) for anticipation by Hite is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1). *See* 37 C.F.R. § 1.136(a)(1)(v) (2010).

Appeal 2009-006328
Application 10/065,793

AFFIRMED

babc

SMITH HOPEN, PA
180 PINE AVENUE NORTH
OLDSMAR, FL 34677